

A Call Control Model for SIP

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#) [[RFC2026](#)].

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts. Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at

<http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at

<http://www.ietf.org/shadow.html>.

1. Abstract

This document defines an abstract call model for describing the media relationships required for call control features in SIP, and discusses other issues related to SIP call control as part of the SIP Call Control Framework.

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC-2119](#) [[RFC2119](#)].

3. Overview

This document defines an abstract call model for describing the changes in media relationships (actions) which are needed to fulfill most call control features in SIP. The call model is an integral

part of the SIP Call Control Framework defined in [[cc-framework](#)]. The model and actions described here are specifically chosen to be independent the SIP signaling and/or mixing approach chosen to actually setup the media relationships.

Mahy

Expires: Jan 2002

1

Call Control Model for SIP

Implementations may setup the media relationships described in this model using the approach described in [[3pcc](#)]. The 3pcc approach relies on only the following 3 primitive operations:

```
new INVITE
reINVITE (modify the session, hold, resume from hold, etc.)
BYE
```

The main advantage of the 3pcc approach is that it only requires very basic SIP support from end systems to support call control features. It also has the advantage and disadvantage that new features can/must be implemented in one place only (the controller), and neither requires enhanced client functionality, nor takes advantage of it.

In addition, a peer-to-peer approach is discussed at length in this draft. The primary drawback of the peer-to-peer model is additional end system complexity. The benefits of the peer-to-peer model include:

- state remains at the edges
- signaling is same whether mixing is performed by one of the participants, or by a central conference server
- calls only go through participants involved (no additional points of failure)
- does not reproduce the MGCP/Megaco command/control trust model
- less complex to setup end-to-end QoS and security
- shorter setup time (fewer messages, and round trips required)
- many additional features applied at other UAs are transparent

The peer-to-peer approach relies on additional "primitive" operations, some of which are identified here.

```
INVITE with [Replaces] semantics
INVITE with Join semantics
INVITE or [REFER] with [Media Forking] semantics
[REFER] to ask another UA to send a request on your behalf
[PHONECTL] for desktop call control
```

Many of the features, primitives, and actions described in this document require some type of mixing/combining/selection.

4. "Conversation Space" Model

This document introduces the concept of an abstract "conversation space" (essentially as a set of participants who believe they are all communicating among one another). Each conversation space contains one or more participants.

Participants are SIP User Agents which send original media to or terminate and receive media from other members of the conversation space. Logically, every participant in the conversation space has

Mahy Expires: Jan 2002 2

Call Control Model for SIP

access to all the media generated in that space (this is true if all participants share a common media type). A SIP User Agent which merely forwards, transcodes, mixes, or selects media originating elsewhere in the conversation space is NOT a participant. [Note that a conversation space consists of one or more SIP calls or SIP conferences. A conversation space is similar to the definition of a "call" in some other call models.]

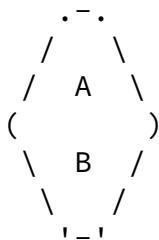
Participants may represent human users or non-human users (referred to as robots or automatons in this document). Some participants may be hidden within a conversation space. Some examples of hidden participants include: robots which generate tones, images, or announcements during a conference to announce users arriving and departing, a human call center supervisor monitoring a conversation between a trainee and a customer, and robots which record media for training or archival purposes.

Participants may also be active or passive. Active participants are expected to be intelligent enough to leave a conversation space when they no longer desire to participate. (An attentive human participant is obviously active.) Some robotic participants (such as a voice messaging system, an instant messaging agent, or a voice dialog system) may be active participants if they can leave the conversation space when there is no human interaction. Other robots (for example our tone generating robot from the previous example) are passive participants. A human participant "on-hold" is passive.

An example diagram of a conversation space shown as a "bubble" or ovals, and as a "set" in curly or square brace notation. Each set, oval, or "bubble" represents a conversation space. Hidden participants are shown in lowercase letters.

{ A , B }

[A , B]



Some examples of the relationship between conversation spaces and SIP calls, SIP call legs, and SIP sessions are listed below. In each example, a human user will perceive that there is a single call.

A simple two-party call is a single conversation space, a single call, a single session, and a single call-leg.

Mahy

Expires: Jan 2002

3

Call Control Model for SIP

A locally mixed three-way call is one or two calls (one if the mixer invited all the other participants, two otherwise), two sessions, and two call-legs. It is also a single conversation space.

A simple dial-in audio conference is a single conversation space, but is represented by as many calls, call-legs, and sessions as there are human participants.

A multicast conference is a single conversation space, a single session, one or more calls, and as many call-legs as participants.

[5. Catalog of call control actions and sample features](#)

Below are listed several call control "actions" which modify the participants in a conversation space. The names of the actions listed are for descriptive purposes only (they are not normative). This list of actions is not meant to be exhaustive.

In the examples, all actions are initiated by the user "Alice" represented by UA "A".

[5.1 Transfer](#)

The conversation space changes as follows:

before after
{ A , B } --> { C , B }

A replaces itself with C.

To make this happen using the peer-to-peer approach, "A" would send two SIP requests. A shorthand for those requests is shown below:

REFER B Refer-To:C
BYE B

To make this happen instead using the 3pcc approach, the controller sends requests represented by the shorthand below:

INVITE C (w/SDP of B)
reINVITE B (w/SDP of C)
BYE A

Features enabled by this action:

- blind transfer
- transfer to a central mixer (some type of conference or forking)
- transfer to park server (park)
- transfer to music on hold or announcement server
- transfer to a "queue"
- transfer to a service (such as Voice Dialogs service)
- transition from local mixer to central mixer

Mahy

Expires: Jan 2002

4

Call Control Model for SIP

[5.2](#) Take

The conversation space changes as follows:

{ B , C } --> { B , A }

A forcibly replaces C with itself. In most uses of this primitive, A is just "un-replacing" itself.

Using the peer-to-peer approach, "A" sends:

INVITE B Replaces: <call leg between B and C>

Using the 3pcc approach (all requests sent from controller)

INVITE A (w/SDP of B)
reINVITE B (w/SDP of A)
BYE C

Features enabled by this action:

- transferee completes an attended transfer
- retrieve from central mixer (not recommended)
- retrieve from music on hold or park

- retrieve from queue
- call center take
- voice portal resuming ownership of a call it originated
- answering-machine style screening (pickup)

[5.3](#) Add

The conversation space changes as follows:

`{ A , B } --> { A, B, C }`

A adds C to the conversation.

Using the peer-to-peer approach, adding a party using local mixing requires no signaling. To transition from a 2-party call or a locally mixed conference to centrally mixing A could send the following requests:

```
REFER B Refer-To: mixer
INVITE mixer
BYE B
```

To add a party to a central mixer:

```
REFER C Refer-To: mixer
or
REFER mixer Refer-To: C
```

Using the 3pcc approach to transition to centrally mixed, the controller would send:

```
INVITE mixer leg 1 (w/SDP of A)
INVITE mixer leg 2 (w/SDP of B)
INVITE C (late SDP)
```

Mahy

Expires: Jan 2002

5

Call Control Model for SIP

```
reINVITE A (w/SDP of mixer leg 1)
reINVITE B (w/SDP of mixer leg 2)
INVITE mixer leg3 (w/SDP of C)
```

To add a party to a central mixer:

```
INVITE C (late SDP)
INVITE mixer (w/SDP of C)
```

Features enabled:

- standard conference feature
- call recording
- answering-machine style screening (screening)

[5.4](#) Local Join

The conversation space changes like this:

`{ A, B } , {A, C} --> {A, B, C}`

or like this

`{ A, B } , {C, D} --> {A, B, C, D}`

A takes two conversation spaces and joins them together into a single space.

Using the peer-to-peer approach, A can mix locally, or REFER the participants of both conversation spaces to the same central mixer (as in 5.3)

For the 3pcc approach, the call flows for inserting participants, and joining and splitting conversation spaces are tedious yet straightforward, so these are left as an exercise for the reader.

Features enabled:

- standard conference feature
- leaving a sidebar to rejoin a larger conference

[5.5](#) Insert

The conversation space changes like this:

`{ B , C } --> {A, B, C }`

A inserts itself into a conversation space.

A proposed mechanism for signaling this using the peer-to-peer approach is to send a new header in an INVITE with "joining" semantics. For example:

`INVITE B Join: <call id of B and C>`

Mahy

Expires: Jan 2002

6

Call Control Model for SIP

If B accepted the INVITE, B would accept responsibility to setup the call legs and mixing necessary (for example: to mix locally or to transfer the participants to a central mixer)

Features enabled:

- barge-in
- call center monitoring
- call recording

[5.6](#) Split

$\{ A, B, C, D \} \rightarrow \{ A, B \} , \{ C, D \}$

If using a central mixer with peer-to-peer

REFER C Refer-To: mixer (new URI)

REFER D Refer-To: mixer (new URI)

BYE C

BYE D

Features enabled:

- sidebar conversations during a larger conference

[5.7](#) Near-fork

A participates in two conversation spaces simultaneously:

$\{ A, B \} \rightarrow \{ B , [A] , C \}$

A is a participant in two conversation spaces such that A sends the same media to both spaces, and renders media from both spaces, presumably by mixing or rendering the media from both. We can define that A is the "anchor" point for both forks, each of which is a separate conversation space.

This action is purely local implementation (it requires no special signaling). Local features such as switching calls between the background and foreground are possible using this media relationship.

[5.8](#) Far fork

The conversation space diagram...

$\{ A, B \} \rightarrow \{ A , [B] , C \}$

A requests B to be the "anchor" of two conversation spaces.

For an example of using 3pcc to setup media forking, see [Media forking]. The session descriptions for forking are quite complex. Controllers should verify that endpoints can handle forked-media, by using some type of Requires header token.

Two ways to setup this media relationship using peer-to-peer call control have been proposed:

- the anchor receives a REFER with require: forked-media (implicit)

- the anchor receives an INVITE with Fork-with header (explicit)

Features enabled:

- barge-in
- voice portal services
- whisper
- hotword detection
- sending DTMF somewhere else

[6. Other Call Control Issues](#)

[6.1 Transparent feature interaction](#)

Combinations of features must work in SIP call control. For example, let us examine the combination of a transfer of a call which is conferenced.

Alice calls Bob. Alice silently "conferences in" her robotic assistant Albert as a hidden party. Bob transfers Alice to Carol. If Bob asks Alice to Replace her leg with a new one to Carol then both Alice and Albert should be communicating with Carol (transparently).

Using the peer-to-peer model, this combination of features works fine if A is doing local mixing (Alice replaces Bob's call-leg with Carol's), or if A is using a central mixer (the mixer replaces Bob's call leg with Carol's). A clever implementation using the 3pcc model can generate similar results.

New extensions to the SIP Call Control Framework should attempt to preserve this property.

[6.2 Presenting information to the user or application](#)

Participants should have access to the names of the other participants in a conversation space, so that this information can be rendered to a human user or processed by an automaton. Although some of this information may be available from To, From, Remote-Party-Id, or other SIP headers, another mechanism of reporting this information may be necessary. [The author believes that the data reported by RTCP is insufficient for these purposes.]

For example, a mixer involved in a conversation space may wish to provide URLs for conference status, and/or conference/floor control.

[6.3 Use of different mixing models](#)

Several conferencing models are discussed in [[conf-models](#)]. For brevity, only the two most popular conferencing models are

Call Control Model for SIP

significantly discussed in this document (local and centralized mixing). Applications of the conversation spaces model to distributed full mesh and multicast conferences are left as an exercise for the reader.

Note that a distributed full mesh conference can be used for basic conferences, but does not allow for more complex conferencing actions like splitting, joining, and forking.

Call control features should be designed to allow a mixer (local or centralized) to decide when to reduce a conference back to a 2-party call, or drop all the participants (for example if only two automotons are communicating).

The actual heuristics used to release calls are beyond the scope of this document, but may depend on properties in the conversation space, such as the number of active, passive, or hidden participants; and the send-only, receive-only, or send-and-receive orientation of various participants.

[6.4](#) Effect when one user is represented by multiple UAs in same call

Multiple participants in the same conversation space may represent the same human user. For example, the user may use one participant for video, chat, and whiteboard media on a PC and another for audio media on a SIP phone. In addition, human users may add robot participants which act on their behalf (for example a call recording service, or a calendar reminder). Call Control features in SIP should continue to function as expected in such an environment.

[6.5](#) "Special" participants

Call control implementation are encouraged to make intelligent decisions based on the type of participants (active/passive, hidden, human/robot) in a conversation space. Currently there is no standard way to convey this information about participants in a conversation space, but work in this area is encouraged.

For example, a music on hold service may take the sensible approach that if there are two or more unhidden participants, it should not provide hold music; or that it will not send hold music to robots.

[6.6](#) Billing issues

Billing in PSTN is typically based on who initiated a call. At the moment billing in a SIP network is neither consistent with itself, nor with the PSTN. (A billing model for SIP should allow for both

PSTN-style billing, and non-PSTN billing.) The example below demonstrates one such inconsistency.

Alice places a call to Bob. Alice then blind transfers Bob to Carol through a PSTN gateway. In current usage of REFER and BYE/Also, Bob may be billed for a call he did not initiate (his UA originated the
Mahy Expires: Jan 2002 9

Call Control Model for SIP

outgoing call leg however). This is not necessarily a terrible thing, but it demonstrates a security concern (Bob must have appropriate local policy to prevent fraud). Also, Alice may wish to pay for Bob's session with Carol. There should be a way to signal this in SIP.

Likewise a Replacement call may maintain the same billing relationship as a Replaced call, so if Alice first calls Carol, then asks Bob to Replace this call, Alice may continue to receive a bill.

Further work in SIP billing should define a way to set or discover the direction of billing.

7. Security Considerations

Let us first examine the security of the primitives used by the 3pcc approach (INVITE, reINVITE, and BYE). All signaling goes through the controller, which is a trusted entity. Initial INVITES are frequently authenticated and may also be hop-by-hop (e.g. IPsec or TLS) or end-to-end (e.g. PGP or S/MIME) encrypted. Also, the human or robot user receiving the INVITE may accept or decline the INVITE based on any number of factors.

An attacker can do many "rude" things to a SIP call-leg today (place calls on hold, send BYEs, reINVITE to a session of their choosing), if they have knowledge of the correct To, From, Call-ID, and CSeq headers. Encrypting or integrity protecting the signaling between User Agents and 3pcc controllers can prevent these attacks.

When using the peer-to-peer approach, the call control actions and primitives are initiated by a) an existing participant in the conversation space, b) a former participant in the conversation space, or c) an entity trusted by one of the participants. For example, a participant always initiates a transfer; a retrieve from Park (a take) is initiated on behalf of a former participant; and a barge-in (insert or far-fork) is initiated by a trusted entity (an operator for example).

Both REFER and PHONECTL primitives can be secured in the same manner as for an initial INVITE. To authorize call control primitives that

trigger special behavior (such as an INVITE with Replace, Join, or Fork semantics), the receiving user agent needs some credentials with which to challenge or authorize the call, as the sender may be completely unknown to the receiver, except through the introduction of a third party. As future work, some form of generic authorization token is probably needed.

8. References

[SIP] M. Handley, E. Schooler, and H. Schulzrinne, "SIP: Session Initiation Protocol", [RFC2543](#), Internet Engineering Task Force, Expires: Jan 2002 10

Call Control Model for SIP

Nov 1998.

[3pcc] J. Rosenberg, J. Peterson, H. Schulzrinne, G. Camarillo, "Third Party Call Control in SIP", Internet Draft <[draft-rosenberg-sip-3pcc-02.txt](#)>, IETF; March 2001. Work in progress

[RFC2026] S Bradner, "The Internet Standards Process -- Revision 3", [RFC2026](#) (BCP), IETF, October 1996.

[RFC2119] S. Bradner, "Key words for use in RFCs to indicate requirement levels," Request for Comments (Best Current Practice) [2119](#), Internet Engineering Task Force, Mar. 1997.

[cc-framework] B. Campbell, "SIP Call Control - Framework ", Internet Draft <[draft-campbell-sip-cc-framework-02.txt](#)>, IETF, Mar. 2001. Work in progress.

[REFER] R. Sparks, "SIP Call Control - Transfer", Internet Draft <[draft-ietf-sip-cc-transfer-04.txt](#)>, IETF; Feb. 2001. Work in progress.

[Replaces] B. Biggs, R. Dean, "The SIP Replaces Header", Internet Draft <[draft-biggs-sip-replaces-00.txt](#)>, IETF, Nov. 2000. Work in progress.

[Media forking] M. Shankar, "SIP Forked Media", Internet Draft <[draft-shankar-sip-forked-media-00.txt](#)>, IETF, Feb. 2001. Work in progress.

[PHONECTL] R. Dean, Belkind, B. Biggs, "PHONECTL: A Protocol for Remote Phone Control", Internet Draft <[draft-dean-phonectl-03.txt](#)>, IETF, Jan. 2001. Work in progress.

[conf-models] J. Rosenberg, H. Schulzrinne, "Models for Multi Party Conferencing in SIP", Internet Draft <[draft-rosenberg-sip-conferencing-models-00.txt](#)>, IETF; Nov. 2000. Work in progress.

10. Acknowledgments

Thanks to all who attended the SIP interim meeting in February 2001 for their support of the ideas behind this document.

11. Author's Addresses

Rohan Mahy
Cisco Systems
170 West Tasman Dr, MS: SJC-21/3/3
Phone: +1 408 526 8570
Email: rohan@cisco.com

Mahy

Expires: Jan 2002

11

Call Control Model for SIP

Full Copyright Statement

"Copyright (C) The Internet Society (date). All Rights Reserved.
This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.
This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

